

## What is SDN?

“Software-defined networking (SDN) is an approach to networking that centralizes control of the network by separating the control logic to off-device compute resources. This enables operators to use programmable control to orchestrate and automate network services without having to physically access the network’s hardware.” – SDNCentral.com

## What is SDN?

“Software-defined networking (SDN) is an approach to networking that centralizes control of the network by separating the control logic to off-device compute resources. **This enables operators to use programmable control to orchestrate and automate network services** without having to physically access the network’s hardware.” – SDNCentral.com



# **INTRO TO NTT GIN SDN**

## Who?

- NTT Global IP Network (AS 2914)
- Started as Verio
- Wholesale IP Transit Network
  - 150+ iBGP Nodes
  - 70+ nodes running full-mesh RSVP-TE
  - 14 Metro-DWDM systems
  - Pseudo-wire Ethernet services available between all nodes
  - Bulk of customer ports are 10GE (or Nx10GE)
  - Present in 42 markets on 5 continents

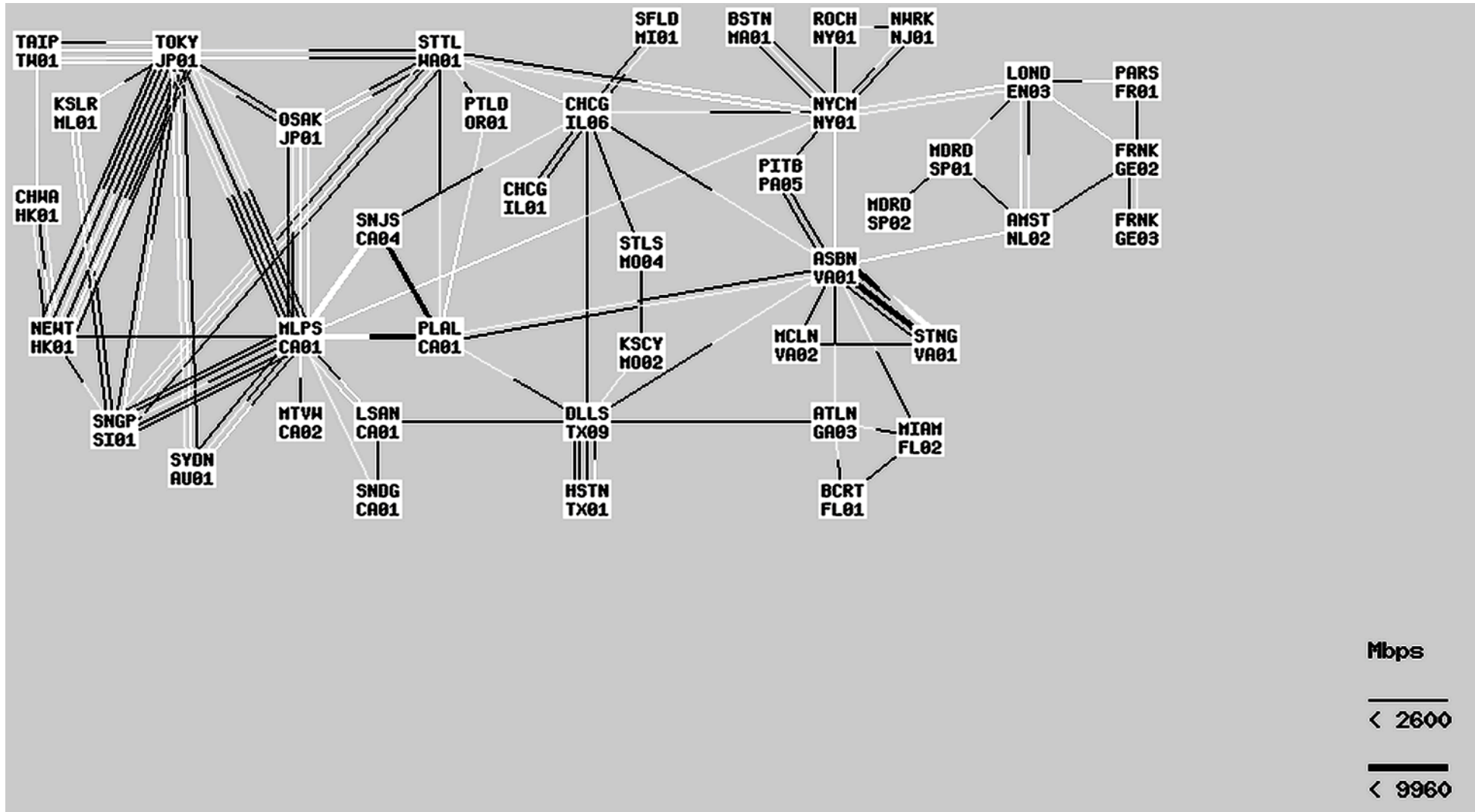
## What?

- GUMS (GIN Unified Management System)
- Fully automated network operation
- Homegrown tools
  - Organic engineering-driven effort
  - Not originally a funded project
  - Development started in late '90s
  - Now employs 4 full-time developers
- Almost to “full” SDN
- Roughly 200 other devices managed by GUMS

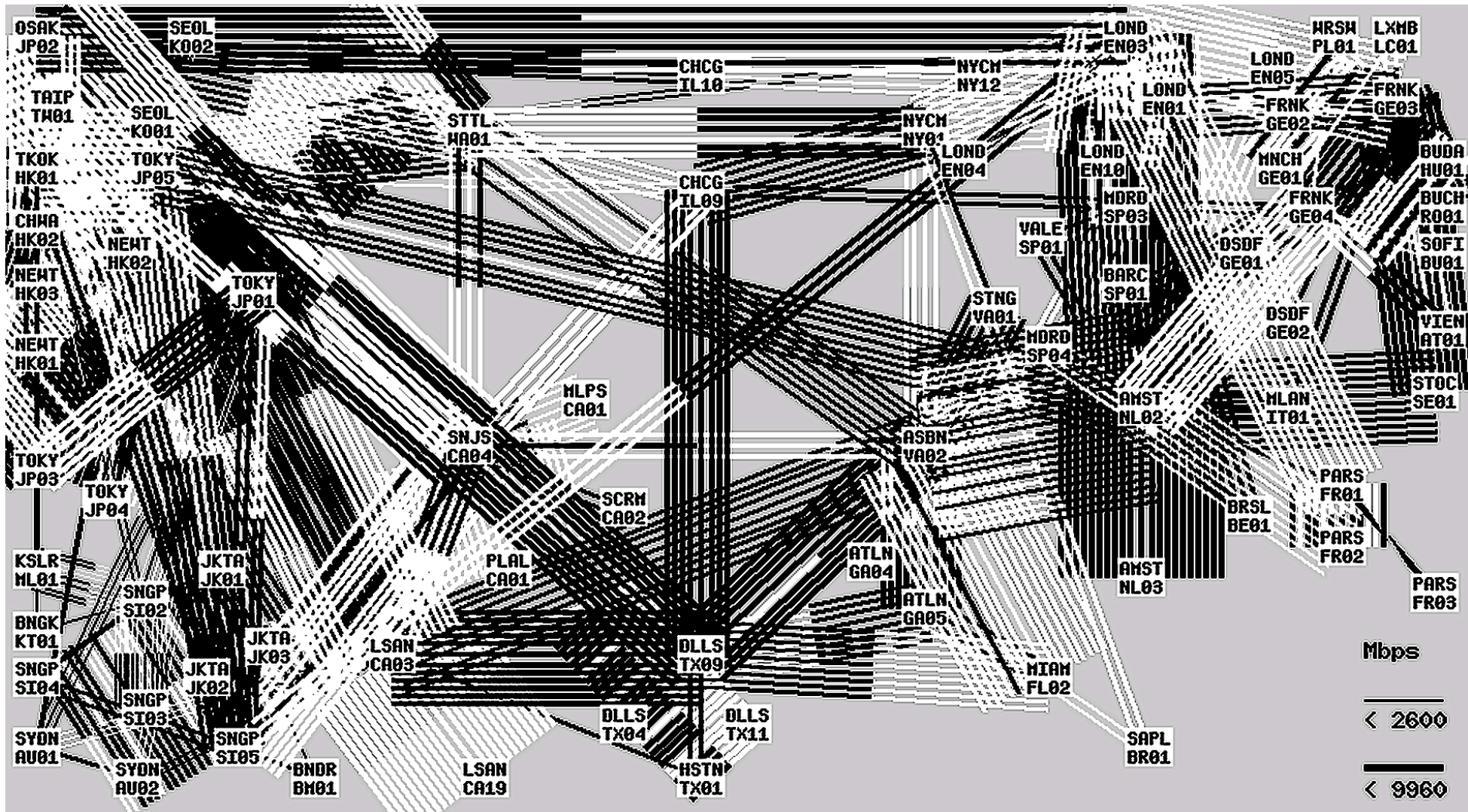
## Why?

- IP Transit pricing experiences a consistent downward pressure
  - Underlying costs must be managed in a similar fashion
- Operating expenses kept low through automating whenever possible
  - Minimize peer review
  - Lower staffing requirements
  - Extensive reporting capabilities
- Higher quality of service
  - Lower error rates (especially catastrophic errors)
  - Consistent service delivery
  - Faster MAC
- Extensive network visibility

GIN 2004.11.01



GIN 2014.11.01





## How?

- Database-driven configuration management system
- Network is modeled in the database
- Data from the database is transformed into device-ready configurations
- Server-side configuration is canonical
  - No persistent manual configuration of devices
- Brute force configuration management

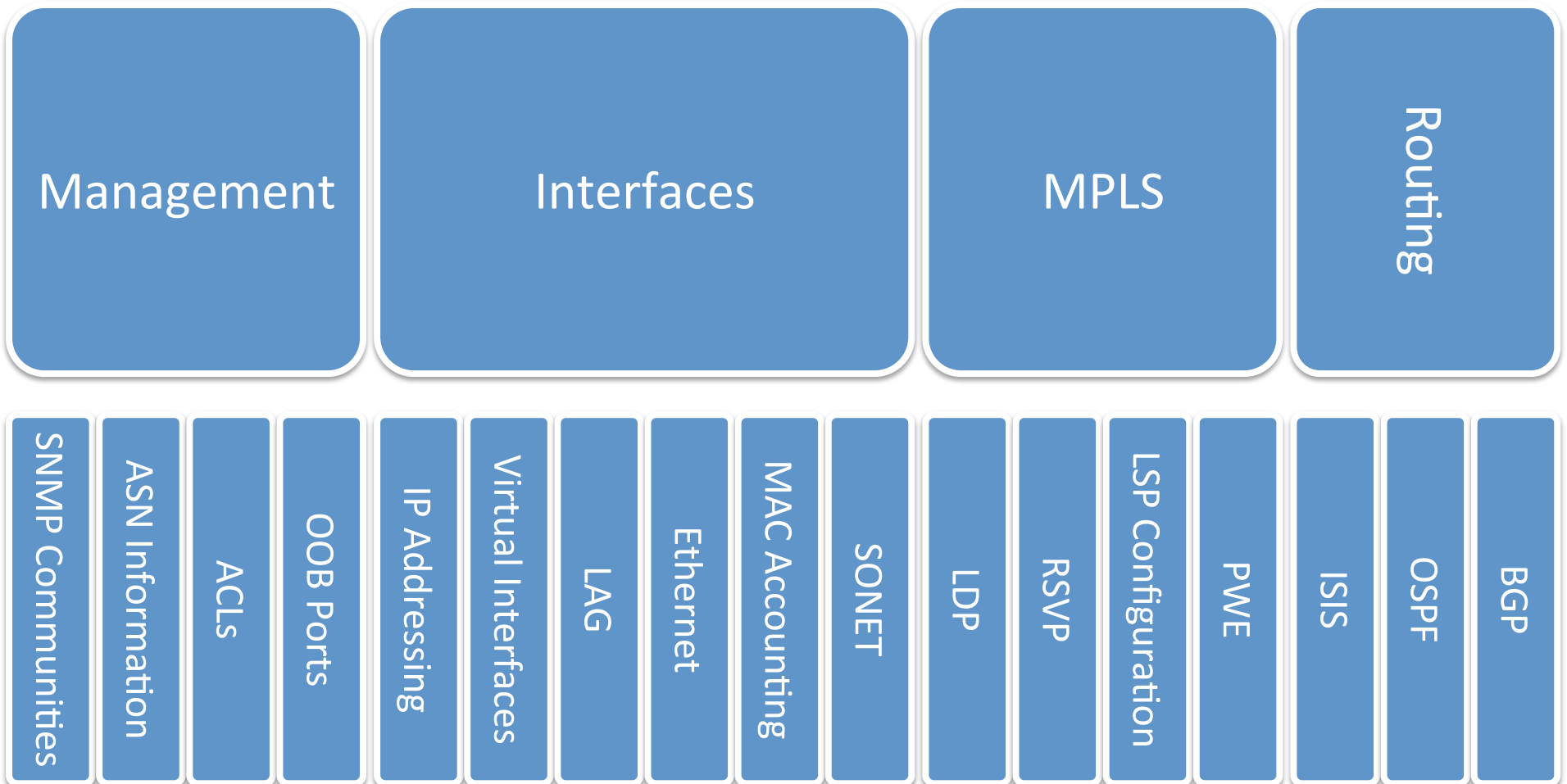
## GUMS Technology

- PostgreSQL
- GNU Make
- M4 Macros
- bgptool homegrown binary
  - Includes customized M4 processor
- Plain text file for each router in CVS
- Custom scripts built on RANCID for pushing configurations to routers
- RANCID collecting configurations hourly
  - Still useful for historical purposes

## What's in the templates?

- Standard device parameters
  - AAA config
  - SNMP
  - Logging
- Interface parameters
- Routing policy
- Can include version-dependent options

## What is in the database?



## What are the router requirements?

- SSH Access
- Ability to retrieve files via FTP
- Commit/roll back/roll forward capability
- If lacking the above, ability to directly manipulate the startup configuration

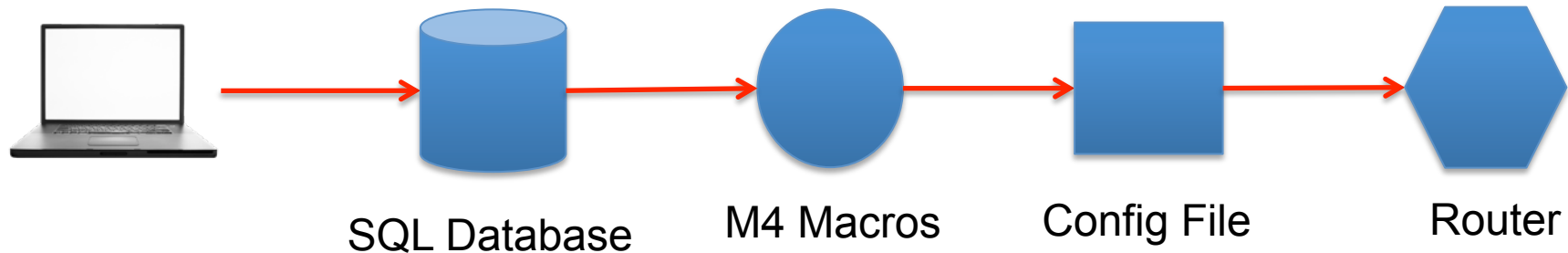
# What's in the plain text file?

```
@DEVICE(myHOST())dn1
PLATFORM(hfr,mcast)dn1
dn1
!
@BANNER(myHOST())dn1
!
SERVICES(`loopback0')dn1
!
dn1 ENABLE()dn1
!
@R_POLICY(myHOST())dn1
!
dn1 NETFLOW must be defined before INTERFACES
_NETFLOW(_COLLECTOR1())dn1
!
@INTERFACES(myHOST())dn1
!
@CLNS(myHOST(), `verio',12,`wide')dn1
!
dn1 @MPLS(myHOST())dn1
!
@STATICS(myHOST())dn1
_BLACKHOLE()dn1
!
@L2VPNU(myHOST())dn1
!
IPEERS(myHOST())dn1
!
@EBGP(myHOST())dn1
```

# What **was** in the plain text file?

```
include(`JNX.m4')dn1
define(`myLOOP',`129.250.0.45')dn1
PLATFORM(juniper,martini)dn1
#
# Verio / PAIX                Palo Alto, CA  Unauthorized Access is Prohibited
# pao6.verio.net              2000.05.17-0  For Service Call  (800) 551-1630
#
@`SERVICES'(myHOST)
SERVICES()dn1
NAMESERVERS()dn1
LOGGING()dn1
USERS()dn1
SNMP(,`PAIX')dn1
#
interfaces {
    so-1/0/1 {
        description "BB: pvu0 p1-0-0-0 - PAIX c34-r4-s3-s-b2b-b3-19-20/MFS o2-brt-u88-0001/Q spa-3003095/ELI oc-obgl-105143-003-
elg";
        clocking external;
        encapsulation cisco-hdlc;
        sonet-options {
            fcs 16;
            payload-scrambler;
        }
        unit 0 {
            point-to-point;
            family inet {
                no-redirects;
                address 129.250.3.25/30;
            }
            COST(13, `so-1/0/1', `BB: pvu0 p1-0-0-0')dn1
            PIMMODE(`sparse-dense', `so-1/0/1.0', 1)
        }
    }
}
```

## GUMS Workflow



1. User enters database changes via Web UI
2. User initiates config build via make command on server
3. User initiates config push via loadconfmem command on server
  1. Router is contacted by script via SSH
  2. Router requests configuration file from server via FTP
  3. Configuration is committed





What kind of applications does this enable?

- Automatic customer BGP ACL and max prefix updates
- BGP configuration tool for IOS
  - All relevant config (ie. Interface, BGP neighbor, policy) is loaded via 'copy ftp: running-config
- Mass update of RSVP-TE LSPs
- Bulk move of interfaces/sub-interfaces
- Seeding of other systems with data (eg. stats, NMS, billing)
- Single interface for complex service instantiation

## Optical SDN

- Using GUMS to provision 10G Optical services
- Using NBI on Cyan's Planet Operate
- Different device interaction model than routers
- Optical equipment companies are mostly clueless when it comes to device management
- Currently no tie between optical and IP service layers

## Challenges

- Need better support for concurrent operations
- Brute force configuration management has limitations
- Most vendors' programmatic configuration solutions are not ready
- Vendors focused on service provisioning
- We want to completely configure the box programmatically



# **WHAT ABOUT MY NETWORK?**

## Does SDN have a place in carrier networks

- Yes!
  - Maybe OpenFlow, PCE, i2rs etc. do too, but not what we're focused on.
  - Routing protocols work fine.
- GUMS is not “full” SDN, yet still realizing tremendous benefits
- Automation is inevitable
- Implementation can be incremental



## What do I need to know?

- There are no magic bullets
- There will be custom development work
- Avoid the “PeopleSoft” problem
- Requires a cultural shift

## Should I build or buy?

- Probably both
- COTS components can be integrated with homegrown approaches
- Both approaches will require development costs
  - Either staffed or outsourced
  - Expertise hard to find either way
- Homegrown provides ultimate flexibility
  - No vendor lock-in
  - No external dependencies for new HW/SW support
- COTS has bigger potential for “PeopleSoft problem”

## Impacts on Organizational and Operational Cultures

- Some groups/employees may feel they are be automated out of a job
  - Automating where possible frees up staff for more rewarding work
- Tight integration between network and development staff makes for the best results
- Systems support can be critical
- Things must still be fixed when they break
- Must remain vigilant for “skill rot”
- Network operators are more effective when they understand the operation of the tools





## Impacts on Equipment Selection

- Integration with SDN toolsets becomes paramount
- Some vendors may willingly or unwillingly remove themselves from contention
- Using COTS may further narrow choices
- Integration new platforms may become easier



## Pitfalls

- Supporting hack solutions may become more difficult
  - Hacks can become landmines
- Costs can quickly spiral out of control if not closely managed
- You can inadvertently give others destructive access to the network
- If not well thought out your system can paint you into a corner



## Conclusions

- Automation is the way forward
  - Remains to be seen whether the concept of SDN will persist
- The tools are getting better everyday
- You can do this!